|  |
| --- |
|  |
| Cahier des charges pour la réalisation du site Scientia |
| Développement et description de la réalisation du site. |

Date de rédaction : 14 décembre 2012

Le client : Yohan Cormier

Contacts technique :

Yohan Cormier, directeur et réalisateur du projet.

7351 rue Saint-Hubert H2R 2N4

Téléphone : 438-993-9898

[Yohan.cormier@gmail.com](mailto:Yohan.cormier@gmail.com)

# Table des matières

Table des matières 2

Objet 3

Notre société 3

Chiffres 3

Notre positionnement concurrentiel 3

Contexte 3

Cibles et objectifs 3

Analyse de l’existant 5

Pilotage 6

Chef de projet 6

Décisionnel métier 6

Fonctionnalités 7

Référencement 7

Contraintes et risques 8

Contraintes techniques 8

Contraintes légales 8

Accessibilité 8

Risques et points de surveillance 9

Méthodologie 10

Planning prévisionnel 10

Annexe : 16

# Objet

## Notre société

N’ayant pas de société dans le présent, il n’y a pas lieu de décrire cette partie.

## Chiffres

La non-entreprise s’élève à 0$ avec un employé bénévole au nom de Yohan Cormier

## Notre positionnement concurrentiel

Nos trois concurrents sont : wikipedia.org, siteduzero.com/sciences.siteduzero.com et academie-en-ligne.fr

L’élément différenciant de notre activité est la volonté d’organiser tout un contenu éducatif à la manière de cours classés par types et niveaux nécessaire pour suivre la formation (ex : Secondaire1, Secondaire2, etc…).

## Contexte

La nécessité de continuer son apprentissage « tout au long de sa vie ». Le besoin est né dans le contexte que je désirais continuer mes études et en même temps de créer un site professionnel pouvant me servir dans mon portfolio.

Ce site web permet de me faire connaitre sur l’Internet et de démontrer mes capacités organisationnel, productive et technique.

## Cibles et objectifs

**CIBLE 1 : Les étudiants.**

* Profil sociodémographique :

Les étudiants francophones (Canada/France/Suisse/Belgique) principalement.

* Besoins/attentes :

Ils ont besoin d’information structurés.

Ils ont besoin d’une méthodologie pour faire des révisions.

* Nature des services que le site leur offre :

Cours entier à disposition.

Révision grâce à des questionnaires.

Gratuité des services.

* Autres moyens de communication actuellement utilisés pour atteindre cette cible :

Rencontre physique, publicité dans les CÉGEP et universités, Facebook.

* Le type d’action que l’on attend d’eux sur le site :

Partage via les réseaux sociaux, inscription, activisme dans le forum, clique sur les bannières de publicités.

**CIBLE 2 : Les professeurs.**

* Profil sociodémographique :

Les professeurs francophones (Canada/France/Suisse/Belgique) principalement.

* Besoins/attentes :

Pouvoir transmettre leur savoir à ceux qui le souhaitent.

Avoir des retours de leurs cours de la part des « étudiants ».

Avoir un accès à des cours préparé pour leurs propres cours (physiquement parlant).

* Nature des services que le site leur offre :

Avoir un accès simple et intuitif pour écrire les cours.

Avoir la possibilité de sélectionner le type de copyright pour le cours (distribuable ou non et de quel façon).

Avoir un moyen de communiquer avec les autres auteurs.

* Autres moyens de communication actuellement utilisés pour atteindre cette cible :

Rencontre physique, publicité dans les CÉGEP et universités, Facebook.

* Le type d’action que l’on attend d’eux sur le site :

Création de cours et assistance auprès de ceux qui en ont besoin.

Partage dans leurs cours (physique) et dans les réseaux sociaux.

Faire de la publicité autour de soi.

**CIBLE 3 : Les utilisateurs d’Internet en général.**

* Profil sociodémographique :

Les utilisateurs d’Internet francophone en général.

* Besoins/attentes :

Cherchent à approfondir leurs connaissances.

Ils ont besoin d’un site qui se maintient sur un sujet et qui soit ordonnées.

Ils ont besoin de personnes pouvant les conseiller.

Ils ont besoin de tester leurs connaissances.

* Nature des services que le site leur offre :

Grande base de données ordonné de façon à ce que chaque sujet soit dans un thème spécifique.

Forum permettant d’émettre une critique sur le cours ou en sujet au cours.

Réseaux social permettant de communiquer avec d’autres inscrits.

Quizz permettant de tester ces connaissances dans un thème ou dans un niveau en général.

* Autres moyens de communication actuellement utilisés pour atteindre cette cible :

Rencontre physique, publicité dans les CÉGEP, collège, universités et Facebook.

* Le type d’action que l’on attend d’eux sur le site :

Inscription au site, activisme dans le forum, teste des quizz de façon ludique, partage dans les réseaux sociaux, clique sur les bannières publicitaires.

## Analyse de l’existant

Lister les outils de communication physiques ou interactifs qui existent déjà : logo et identité visuelle, plaquettes, courriers, prospectus, e-mails avec nom de domaine, affichage, PLV, site Web, intranet, extranet, blog, newsletter, compte sur les réseaux sociaux, etc.

Définir les outils à conserver ou supprimer.

Comment le nouveau site s’interfacera-t-il avec cet existant ?

Faut-il assurer la reprise des données dans le projet ?

Le site du zéro (siteduzero.com)

Wikipedia (wikipedia.org)

# Pilotage

## Chef de projet

L’interlocuteur sur ce projet sera Yohan Cormier. Il aura pour rôle de :

* Définir et rédiger (ou valider) les besoins
* Définir le planning, le budget et s’assurer qu’ils sont respectés
* Faire l’interface entre les clients et le pilotage technique

## Décisionnel métier

Des phases de validation impliqueront différents intervenants au sein de notre structure :

* Validation des contenus : directeur de la communication ?
* Validation graphique : directeur artistique ?
* Validation technique : Yohan Cormier
* Tests fonctionnels/recette : équipe communication ?Prestations attendues

## Fonctionnalités

Fonction de gestion de compte des utilisateurs.

Fonction contrôlant l’affichage.

Fonction de contrôle de l’accessibilité.

Fonction de réécriture d’URL.

Fonction permettant d’écrire pour un auteur de manière structuré et adaptatif au type de cours.

Fonction de sitemap.

Fonction de fil d’Ariane.

Fonction de système d’audit (système, DB, etc…).

Fonction de quizz.

Fonction de recherche.

Fonction d’inscription des utilisateurs.

Fonction de forum de discussion.

Fonction de communication et de partage avec les réseaux sociaux.

Fonction d’obtention du contenu selon les droits qui y sont liés.

Fonction de statistique (compteur, etc…).

Fonction de système de cache.

Fonction de sondage.

Fonction de calendrier.

## Référencement

Le site devra intégrer les bonnes pratiques en matière de SEO (optimisation pour les moteurs de recherche) :

* balisage sémantique
* adresses de pages lisibles (URL rewriting)
* navigation sans Javascript/Flash
* présence sur les réseaux sociaux (Facebook, Twitter, Linkedin, etc.)
* présence dans les annuaires web.
* inscription Google Adresses
* tenue d’un blog professionnel
* participation à des forums
* achat de mots-clés (Google Adwords, Microsoft, etc.)
* plateforme d’affiliation
* analyse concurrentielle du référencement (où sont vos concurrents ?)

# Contraintes et risques

## Contraintes techniques

Les langages de développement seront HTML5, CSS3, Javascript, PHP (éventuellement évoluant en python une fois le langage maitrisé) et SQL.

Le serveur devra, au minimum, être un système de type Unix/Linux avec un serveur Apache2.2 et les modules suivant :

* Python
  + Django
* MySQL 5.0.x
* phpMyAdmin 3.3.10.x
* PHP 5.3.1
  + PDO
  + PEAR
  + GD
* Mod rewrite rules d’Apache

Le Framework sera axé sur les sur les trois points suivant :

* La sécurité. Chaque donnés envoyé et reçus auras été vérifié.

La sécurité sera pensée pendant tout le processus de la conception et s’écoulera sur les deux autres points. Il devra être considéré comme l’un des points le plus important du projet de la conception du Framework. Sans avoir effectué une batterie de test et d’avoir été approuvé par une majorité, l’application ne pourra faire partie du Framework.

* Le dynamisme/l’adaptation. Il sera facile d’améliorer ou de modifier une petite ou grande partie du site.
* L’administration/le contrôle. Contrôle total sur le site. Administration contrôlé par une série de type de compte tous adapté à une fonction donné.

## Contraintes légales

Voir si besoin d’une déclaration de type CNIL.

Exemple :

[Délibération n°2005-284 du 22/11/2005 décidant la dispense de déclaration des sites web diffusant ou collectant des données à caractère personnel mis en oeuvre par des particuliers dans le cadre d'une activité exclusivement personnelle.](http://www.legifrance.gouv.fr/jopdf/common/jo_pdf.jsp?numJO=0&dateJO=20051217&numTexte=79&pageDebut=&pageFin=)

(http://www.legifrance.gouv.fr/jopdf/common/jo\_pdf.jsp?numJO=0&dateJO=20051217&numTexte=79&pageDebut=&pageFin=)

## Accessibilité

Exigence relativement importante en matière d’accessibilité aux publics handicapés :

* L’utilisation des standards du Web (HTML), la hiérarchisation stricte de l’information selon les balises sémantiques et la séparation entre le contenu et la forme (usage des CSS) garantissent une accessibilité optimale sur le plus grand nombre de navigateurs et de dispositifs de navigation.
* Les systèmes de script dynamiques (Javascript) pourront être exploités pour des fonctionnalités enrichies mais leur usage n’est pas indispensable à l’utilisation du site
* Le projet n’emploie pas la technologie Flash dans son fonctionnement normal

## Risques et points de surveillance

* Données sensibles : prévoir une anonymisation, cryptage et une obfusation.

## Méthodologie

Le développement choisi sera en cascade à partir d’un cahier des charges complet.

Une partie du développement se fera en développement par versions successives de façon à permettre aux auteurs de commencer la production pendant le développement.

## Planning prévisionnel

Note :

* les principales étapes de conception-réalisation qui découpent le projet (prévoir un diagramme de PERT)
* les éventuelles « versions » si l’ensemble des fonctionnalités ne peut être intégré en une étape
* la phase de recette (phase de test et de vérification de l’outil avant mise en ligne)
* les réunions et phases de validation (en précisant les phases de recette qui solliciteront nos équipes)

Phase première :

Développement du Framework de travail.

Définition du Framework :

Le Framework utilisera le moins possible d’éléments extérieure pour être construit et devra répondre à une norme de sécurité, de dynamisme et d’adaptabilité élevé (description plus poussé en-dessous et dans contraintes techniques).

La sécurité :

La sécurité est un des éléments le plus important du présent Framework, il sera développé et pris en compte au niveau de la programmation, de la structure et de l’environnement du site.

Le dynamisme :

Le code devra être arrangé de manière telle qu’il sera possible de changer une partie du code sans pour autant avoir à changer le reste.

L’adaptation :

Le Framework sera permissif concernant la construction structurelle du site et permettra nativement son propre choix de moteurs de Template et de ces bibliothèques.

Schéma hiérarchique de la structure du site et description.

/

.htaccess

admin/

.htaccess

index.php

admin.php

config/

root/

tmp/

public/

index.php

media/

js/

addons/

libs/

css/

screen/

print/

others/

fonts/

imgs/

icons/

home/

.htaccess

var/

.htaccess

sessions/

cache/

tmp/

views\_compiles/

logs/

report.log

contributors\_report.log

errors.log

warnings.log

alerts.log

uploads/

lock/

tmp/

.htaccess

libs/

.htaccess

config/

.htaccess

app/

.htaccess

functions/

default/

configs/

models/

modules/

(module 1)/

config/

controller/

model/

view/

components/

(component 1)/

config/

controller/

model/

view/

plugins/

(plugin 1)/

config/

controller/

model/

view/

scripts/

.htaccess

docs/

.htaccess

sandbox/

La racine du site (/) obligera l’utilisateur à aller dans le dossier « public » du site pour y protéger l’accès aux dossiers et fichiers dit sensible.

**/admin/ -** Dossier principal pour l’administration technique du site.

**/admin/index.php –** Fichier principal du dossier qui sécurise l’accès et le type de demande selon les droits qui seront envoyés au fichier admin.php.

**/admin/admin.php –** Fichier principal de la gestion de l’administration. Gère l’accès et le type de demande selon le droit de l’utilisateur.

**/admin/config/ -** Dossier contenant les configurations en rapport avec l’administration du site.

**/admin/root/ -** Dossier de l’administrateur principal du site.

**/admin/root/tmp/ -** Dossier contenant des données temporaire ou variable c-a-d qui change fréquement.

**/public/ -** Seul dossier a accès dit public avec du contenu pas confidentiel.

**/public/index.php –** Fichier principal du site entier. Sa fonction principal est de diriger les séquences d’inclusion primaire.

**/public/media/ -** Contient tout ce qui se réfère au visuel. Contient ce qui parait du coté client.

**/public/media/js/ -** Dossier contenant les scripts JavaScript.

**/public/media/js/addons/ -** Dossier contenant les extensions JavaScript.

**/public/media/js/libs/ -** Dossier contenant les bibliothèques JavaScript.

**/public/media/css/ -** Dossier contenant les scripts css.

**/public/media/css/screen/ -** Dossier contenant les scripts css adapté pour les écrans.

**/public/media/css/print/ -** Dossier contenant les scripts css adapté pour imprimer.

**/public/media/css/others/ -** Dossier contenant les scripts css pour d’autre formats.

**/public/media/css/fonts/ -** Dossier contenant des polices d’écriture.

**/public/media/imgs/ -** Dossier contenant les images du site.

**/public/media/icons/ -** Dossier contenant les icônes du site.

**/home/ -** Dossier pour chaque utilisateurs participant à la réalisation du site.

**/var/ -** Ce répertoire contient des fichiers qui peuvent changer régulièrement comme les fichiers de journalisation.

**/var/sessions/ -** Dossier contenant les sessions ouverte en cour.

**/var/cache/ -** Dossier contenant des fichiers temporaire de code ré employable.

**/var/tmp/ -** Dossier pour le téléchargement et le chargement des fichiers des participant au site.

**/var/views\_compiles/ -** Dossier contenant les pages du site les plus demandé en pré-généré.

**/var/logs/ -** Dossier contenant le système journalisation.

**/var/logs/report.log –** Rapport général du site.

**/var/contributors\_report.log –** Rapport des utilisateurs participant à la réalisation du site.

**/var/errors.log –** Fichier de rapport des erreurs.

**/var/warnings.log –** Fichier de rapport des avertissements.

**/var/alerts.log –** Fichier de rapport des alertes.

**/var/uploads/ -** Dossier contenant les uploads temporaire. Chaque entrée dans le site doit être confirmée au minimum par une autre personne que celui qui l’a uploadé.

**/var/lock/ -** Contient les fichiers de verrouillage.

**/tmp/ -** Dossier pour le téléchargement et le chargement de fichier pour l'utilisateur.

**/libs/ -** Contient les bibliothèques telles que Zend Framework et les moteurs de templates tel que Smarty.

**/config/ -** Dossier contenant les configurations du site Internet (Nom du site, adresse de la base de données, etc…).

**/app/ -** Contient les applications du site.

**/app/functions/ -** Dossier contenant des functions.

**/app/default/ -** Dossier appelant les applications minimum nécessaire pour le fonctionnement du site avec les fichiers de configuration de base.

**/app/configs/ -** Fichiers contenant les configurations de base et la liste des applications de démarrage.

**/app/models/ -** Contient la configuration des modèles pour le chargement par défaut.

**/app/modules/ -** Répertoire des modules.

**/app/modules/(module 1)/ -** Répertoire du module 1. Un répertoire par module.

**/app/modules/(module 1)/config/ -** Contient les fichiers de configuration du module.

**/app/modules/(module 1)/controller/ -** Contient les fichiers du contrôleur du module.

**/app/modules/(module 1)/model/ -** Contient les fichiers du model du module.

**/app/modules/(module 1)/view/ -** Contient les vues du module.

**/app/components/ -** Répertoire des composants.

**/app/components/(component 1)/ -** Répertoire du composant 1. Un répertoire par composant.

**/app/components/(component 1)/config/ -** Contient les fichiers des configuration du composant.

**/app/components/(compenent 1)/controller/ -** Contient les fichiers du contrôleur du composant.

**/app/components/(component 1)/model/ -** Contient les fichiers du model du composant.

**/app/components/(component 1)/view/ -** Contient les vues du composant.

**/app/plugins/ -** Répertoire des plugins.

**/app/plugins/(plugin 1)/ -** Répertoire du plugin 1. Un répertoire par plugin.

**/app/plugins/(plugin 1)/config/ -** Contient les fichiers de configuration du plugin.

**/app/plugins/(plugin 1)/controller/ -** Contient les fichiers du contrôleur du plugin.

**/app/plugins/(plugin 1)/model/ -** Contient les fichiers du model du plugin.

**/app/plugins/(plugin 1)/view/ -** Contient les vues du plugin.

**/scripts/ -** Répertoire de script de construction et de maintenance (système cron, etc…).

**/docs/ -** Dossier de la documentation du site et des applications après génération.

**/sandbox/ -** Dossier permettant de tester des applications avant de les mettre en production (mode développement).

Programmation du système de base.

La première étape devra être développée en version successive pour permettre par la suite la production du contenu par les participants.

La seconde partie seras développé en cascade pour pouvoir mettre en ligne le site de façon rapide.

Étapes de programmation du système de base en version successive.

1. Conception et configuration de la base de données.
2. Installation du système de configuration.
3. Installation et configuration de la gestion des comptes.
4. Installation de la fonction de contrôle de l’accessibilité.
5. Installation de la fonction de contrôle du htaccess tel que la réécriture d’URL.
6. Installation d’un WYSIWYG adapté au type de cours.

Étapes de programmation du système en cascade

1. Installation d’un sitemap.
2. Installation d’un fil d’Ariane.
3. Installation d’un système d’audit (système, DB, etc…).
4. Installation d’un système de quizz.
5. Installation d’un système de recherche.
6. Installation d’une fonction de gestion des utilisateurs (inscription pour forum, etc...).
7. Installation d’un forum de discussion.
8. Installation d’une fonction de partage avec les réseaux sociaux.
9. Installation d’un gestionnaire pour l’impression de contenu selon les droits qui y sont attribué.
10. Installation d’un système de statistique du site (page les plus vue, utilisation de la DB, etc..).

Mise en ligne du site.

1. Installation du système de cache.
2. Installation d’un système de sondage.
3. Installation d’une application de calendrier.

Phase deuxième :

Phase de recette.

La phase de test pour la vérification du respect des normes inscrit dans le cahier des charges ci présent, devra être réalisée avant la mise ne ligne et avant l’étape de la programmation en cascade pour ne pas avoir à corriger aussi le contenu entré par les participants.

La phase de test devra respecter en particulier les normes de sécurité et d’accessibilité mentionnée plus haut (contraintes techniques)

Pour passer la phase de test, il faudra avoir été approuvé par la totalité des membres participants à sa conception et aux tests.

La phase de test s’étalonne sur trois étapes :

* La vérification du respect des normes décrites dans ce cahier (syntaxe, accessibilité, respect du formatage du code, respect de la charte graphique, etc…).
* La vérification de la stabilité du site (tentative de piratage, vérification de la réaction face à un évènement imprévu, etc…).
* Les fonctionnalités du site.

# Annexe :